**Rapport Projet Machine Learning**

*BENIDDIR Alicia*

*GOULAHSEN Elisa-Souad*

*MAVOULANA Moubina*

*M2LSI1APP*

**SOMMAIRE :**

[**I. La description du projet :**](#_bfwzayvnyjdu) **1**

[**II. Les étapes de traitement de données :**](#_h5h60b74ttim) **2**

[**III. Gestion du cycle de vie de projet machine learning avec MLFLOW :**](#_mao4tjx7j9eu) **6**

[**IV. Les shap values :**](#_gadbs5l8r54m) **7**

[**V. Conclusion :**](#_ypdm1yj9mjgi) **9**

## **La description du projet :**

Le but de ce projet est d'entraîner un modèle de Machine Learning pour “Home Credit Risk Classification”, qui pourra prédire si un individu a des difficultés, ou pas, à rembourser un prêt. Une fois le modèle entraîné, il faut l’enregistrer et gérer les étapes du cycle du projet ML via l’outil MLFlow. La librairie Shap nous permettra également de rentrer dans l’interprétabilité des prédictions réalisées par le modèle, grâce au concept des Shap Values.

Le data set utilisé provient de Kaggle:https://www.kaggle.com/c/home-credit-default-risk/data, seulement les fichiers application\_train.csv et application\_test.csv ont été traités.

## **Les étapes de traitement de données :**

L'entraînement de notre algorithme de prédiction a nécessité un traitement approfondi des données. Ce traitement a été divisé en plusieurs étapes :

* ***Data Collection  ou la collecte de données :***

La collecte de données consiste à récupérer les données brutes, et à les exporter en local sous un format exploitable. Nous avons donc récupéré les données en lisant des fichiers csv.



* ***Data Analysis ou l’analyse de données :***

Cette analyse est faite en amont des modifications des données brutes. Afin de savoir quelles modifications appliquer, il faut d’abord analyser les données. Nous avons donc étudié les différentes features de notre CSV. On analyse notre dataset. Il s'agit de connaître les types de chaque features qui constituent notre jeu de données il peut s’agir d’un bon indicateur, ou complément d’informations pour la suite de l’analyse. Ceci permet également de comprendre des erreurs de code plus rapidement, si on connaît les types des features récupérées par le code python.

Il est possible de réaliser de nombreuses analyses de données, grâce à de nombreux outils (notamment des outils statistiques). Dans le cadre de notre projet, nous avons procédé à l’analyse d’un seul aspect de notre dataset, afin de rester dans un modèle simplifié du cycle de vie du projet de Machine Learning. En effet, l’aspect que nous avons analysé a été les proportions d’exemples de données que nous possédions dans notre dataset : En récupérant les données brutes, on a pu voir que le nombre de lignes dont la prédiction était à 0 était très disproportionné par rapport à celui dont la prédiction était de 1. Cette analyse débouche alors sur des traitements dans les étapes de cleaning et de feature engineering, permettant d’avoir un dataset plus équilibré, ce qui conduira donc à un meilleur apprentissage de notre modèle.

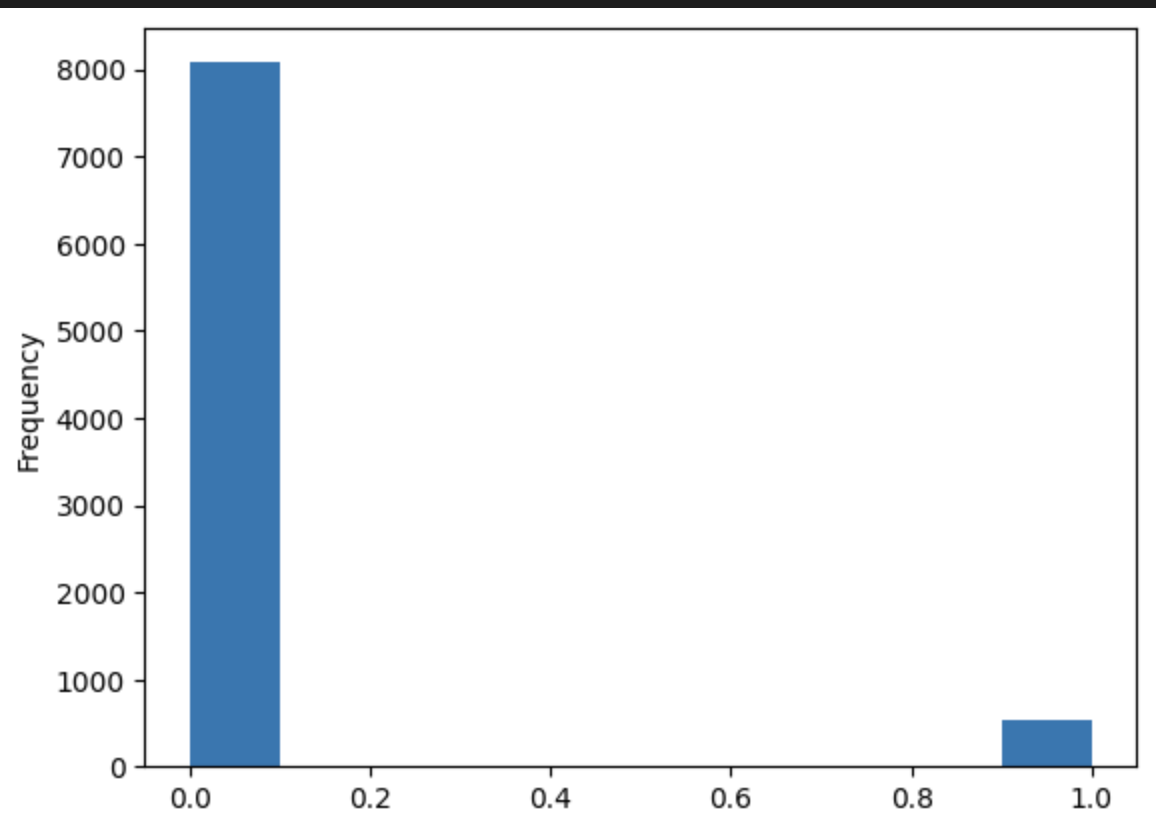
* ***Data Cleaning ou Nettoyage des données***

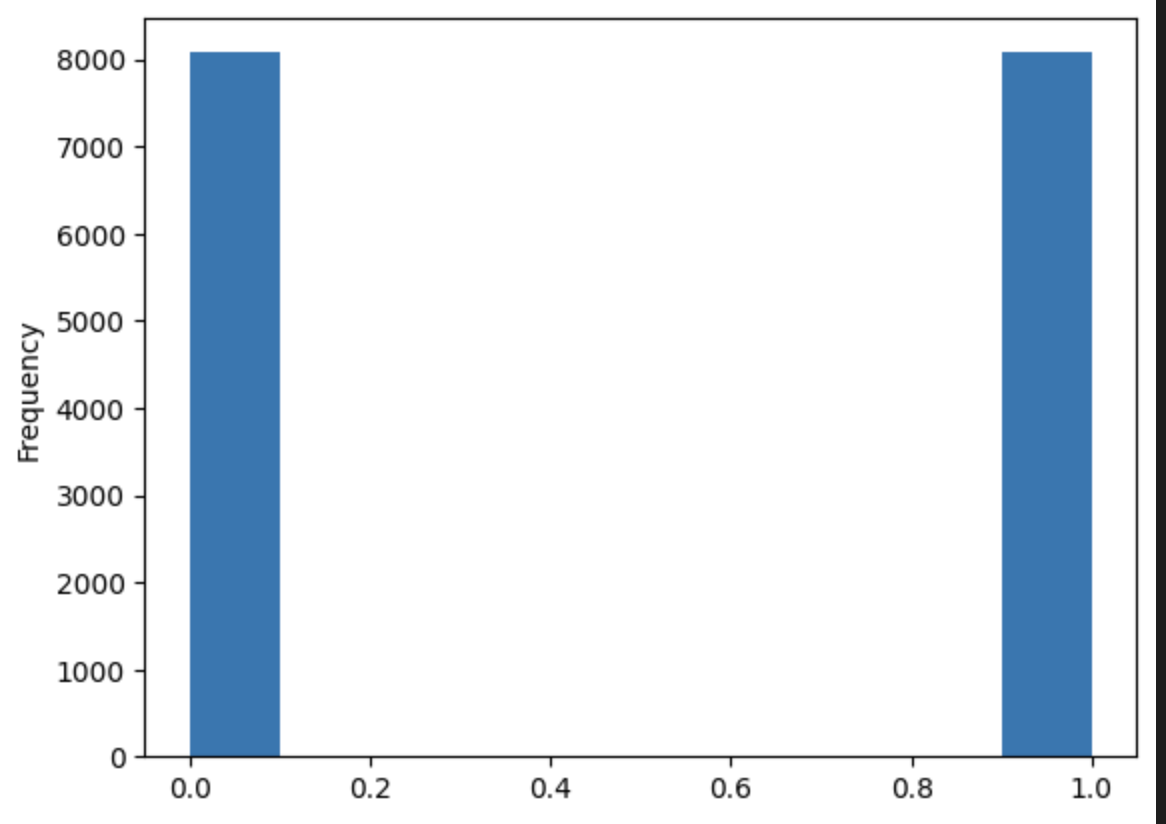
Cette étape consiste à nettoyer le dataset en différentes sous-étapes, et en fonction du dataset que nous avons. Il est possible d’avoir à modifier certains types de features, ou certains caractères spéciaux du dataset. Suite à l’analyse, nous aurions pu également supprimer certaines colonnes qui n’ont pas d’impacts.

Dans un ensemble de données brutes, certaines données peuvent être incomplètes, on parle alors de valeurs manquantes. Dans le but d’avoir un ensemble de données fiable et pertinent, nous avons supprimé toutes les valeurs manquantes de notre data set.

* ***Feature Engineering***

Cette étape permet de créer de nouvelles données à partir des données brutes de notre dataset. Ces nouvelles données permettent un meilleur apprentissage de notre modèle, et des prédictions plus fiables. Dans notre cas, suite à l’analyse de notre dataset déséquilibré, nous avons fait de l’augmentation de données, pour générer d'avantages d’exemples de la classe dont le nombre d’exemples était moindre dans le dataset.





* ***Data Preprocessing***

Cette étape permet de faire une mise à l’échelle des valeurs, ce qu’on appelle du « scaling ». Elle permet de standardiser les valeurs des features numériques pour que celles-ci soient plus compréhensibles par le modèle à entraîner. Il permet d’encoder les valeurs des features catégorielles (qualitatives).

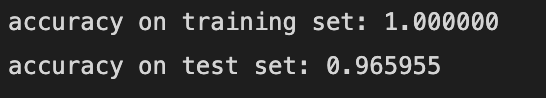
Pour ce projet, nous avons utilisé le label Encoding. On va remplacer chaque valeur possible de la variable par un nombre. Nous l’avons utilisé parce que l’algorithme du random forest ne comprenait pas les strings. Il fallait donc les traduire en valeur numérique pour qu’il puisse les interpréter.

* ***Entrainement du modèle***

Cette étape consiste à entraîner le modèle souhaité avec les données traitées et l’algorithme choisi. Après avoir traité les données, nous avons décidé d’utiliser le random forest comme algorithme pour faire l'entraînement du modèle.

* ***Evaluation des performances***

Cette étape est la dernière du cycle avant la mise en production. Elle permet d’évaluer, par le biais de différentes métriques, les résultats des prédictions de notre modèle : Elle nous permet ainsi de mesurer la fiabilité des prédictions réalisées par notre modèle de Machine Learning. Cette étape finit toujours par nous faire repasser sur les étapes antérieures afin d’ajuster les différents hyperparamètres et/ou le traitement fait sur les données, afin d’affiner la précision de notre prédiction.



## **Gestion du cycle de vie de projet machine learning avec MLFLOW :**

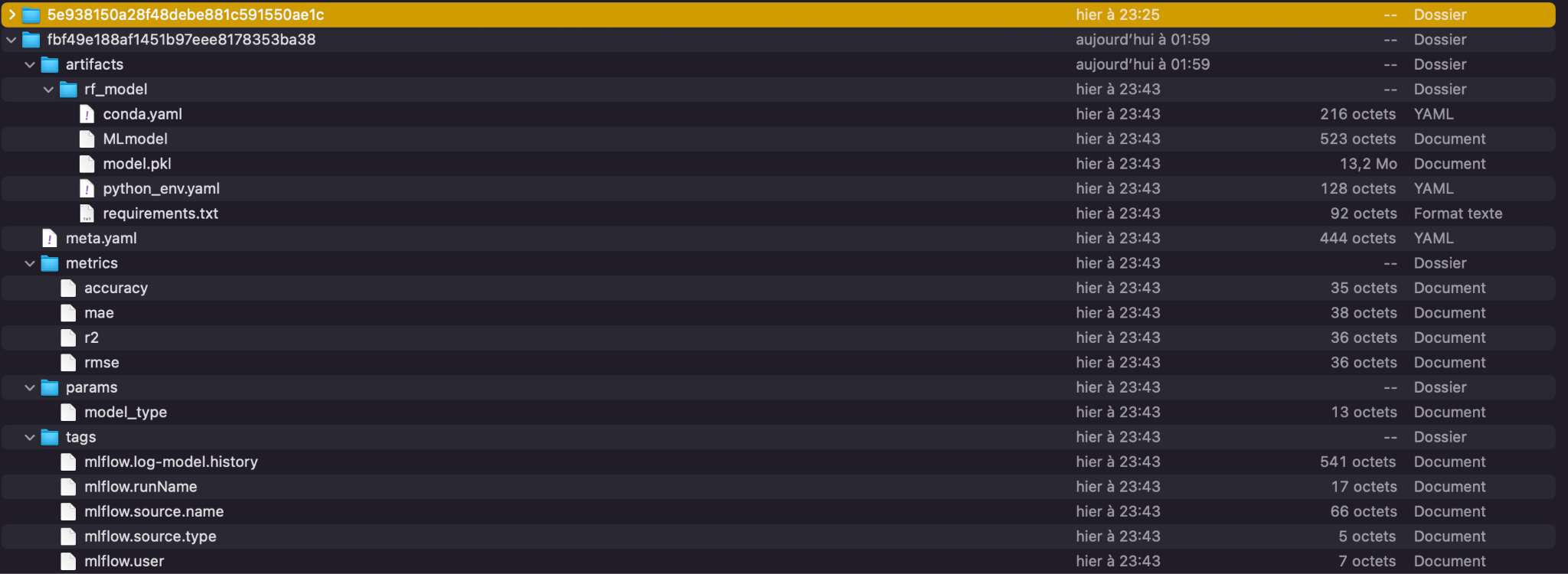
ML Flow est un outil de gestion de cycle de vie de projet machine learning qui permet de gérer et de suivre les expériences de modèles. Ses trois buts sont d’enregistrer et de requêter les expériences comme le code, les données ou les résultats par MLflow tracking. Aussi, il permet de packager pour rendre les expériences productives avec MLflow Projects. Enfin, il permet de proposer un format générique pour envoyer les modèles vers plusieurs outils de déploiements avec MLflow Models.

L’interface de mlflow permet de visualiser et d’enregistrer des données de machine learning. Il permet d’enregistrer des métriques et des modèles et permet de déployer en production. Tout d’abord, pour installer mlflow, il suffit d’avoir installer une version de python et de faire pip install mlflow.

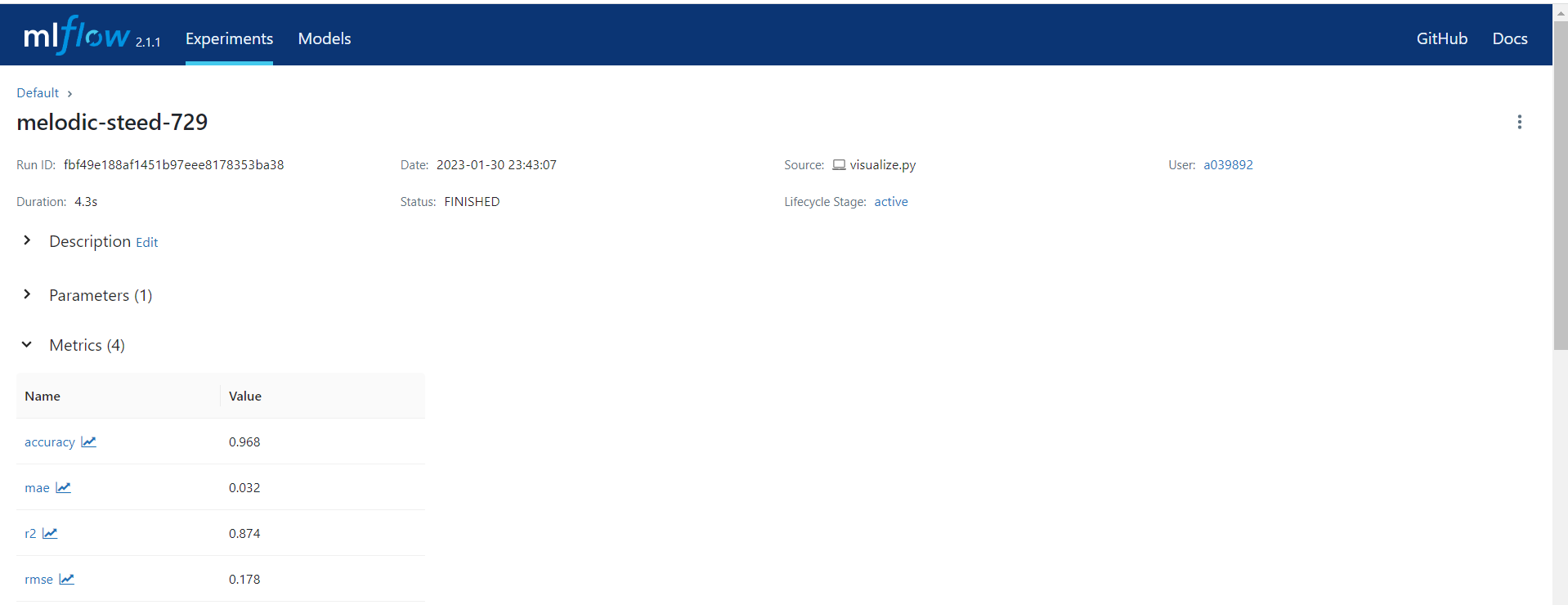
Pour enregistrer des informations, on utilise des loggers. Par exemple, pour enregistrer des résultats, on utilise mlflow.log\_metric.

On crée un fichier train.py qui permet de construire et d’ntraîner le modèle de machine learning. Lors de l'entraînement, le script construit le modèle en utilisant l’algorithme du random forest et entraîne ce modèle en utilisant les données d'entraînement. Les métriques lui permettent d’évaluer les performances du modèle comme l’erreur de prédiction.

Pour visualiser les résultats, on va utiliser mlflow ui puis ouvrir le localhost:5000. On peut voir les modèles enregistrés et les entrées et sorties dans la plateforme, ce qui permet de retracer les différentes itérations d’un modèle pour pouvoir comparer les performances. On peut y apercevoir tous les entraînements avec les jeux de paramètres qui peuvent être différents ou similaires. Lorsque MLFlow est lancé, un dossier mlruns est créé avec un dossier experiments qui contient les informations de chaque experiences, le dossier 0. A l’intérieur sont répertoriés chaque lancé de “python run train.py” avec tous les entraînements. Dans le dossier experiment 0, il y a les informations de chaque runs avec un id. Dans chaque entrainement, il y a un dossier rf\_model qui est le random forest, c'est-à-dire l’algorithme utilisé pour l'entraînement du modèle. Il y a également un dossier metrics qui contient les valeurs sorties par l'entraînement du modèle et les params qui sont les paramètres en entrée. Il y a un fichier meta.yaml dans le dossier 0, ou il y a toutes les informations de l'expérience et contient toutes les informations visibles au niveau du run. On peut ensuite déployer les modèles sur des serveurs ou des conteneurs pour utiliser en production.



On peut voir les paramètres et les métriques avec leur valeurs.



## **Les shap values :**

Le concept des shap values est tiré de celui des valeurs shapley, qui est d’expliquer la contribution de chaque feature dans une prédiction réalisée, plus communément appelée output.

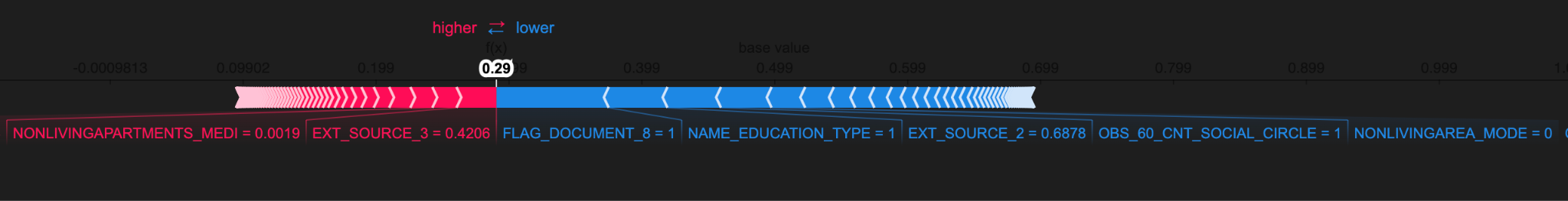
La prédiction consiste à prendre en entrée plusieurs caractéristiques (features), et à en sortir un nombre.

Ainsi, les shap values vont tenter d’expliquer des modèles de Machine Learning, plus ou moins complexes, en mettant en place des modèles simples, qui reproduiront les mêmes résultats.

Pour créer ces modèles simples, les shap values vont utiliser des fonctions linéaires sur les features, et associer chacune d’elles à une valeur réelle. Cette dernière représentera ainsi l’importance de la feature dans la prédiction.

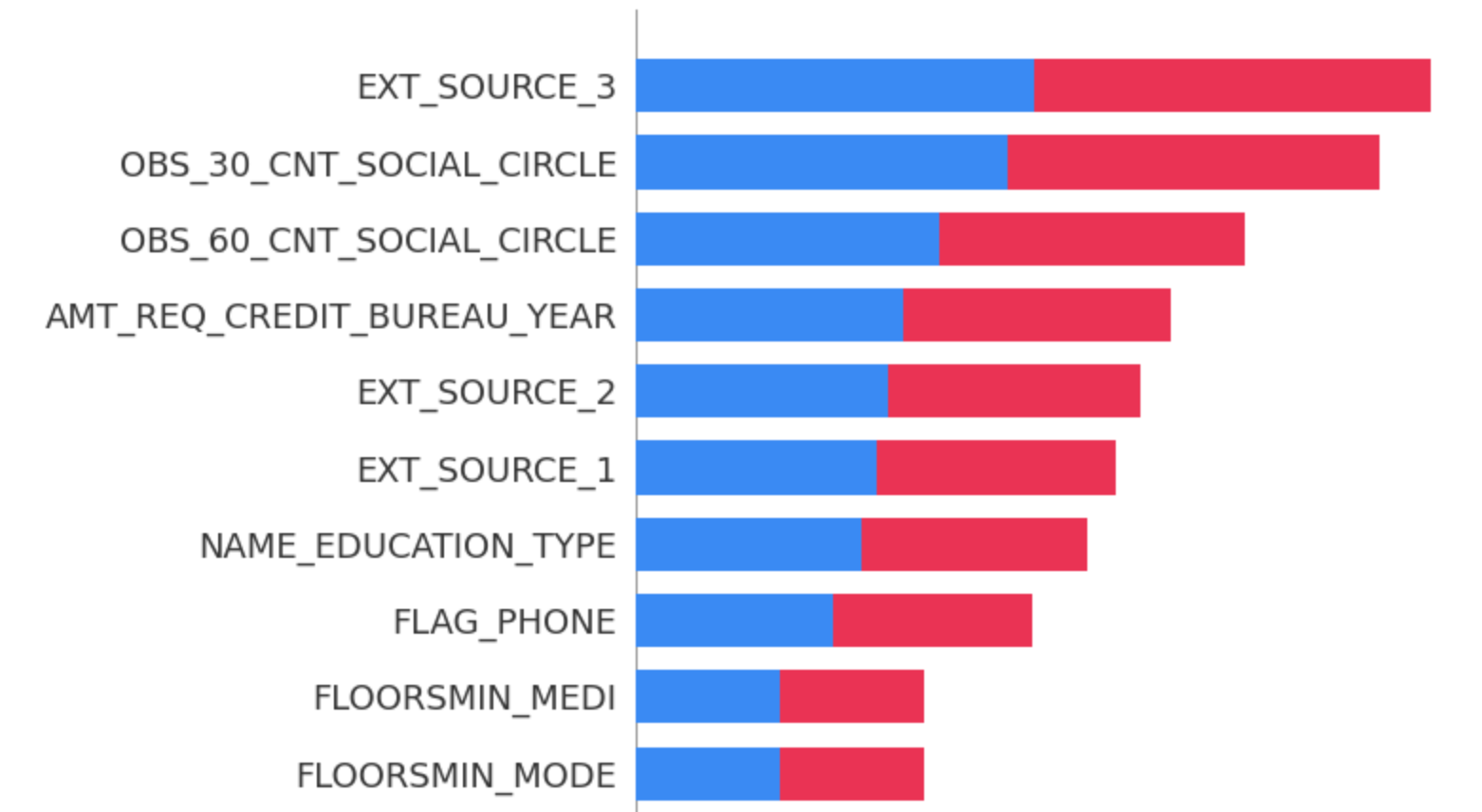
Cette valeur permettra également d’évaluer d’autres paramètres tels que, entre autres, l’influence positive ou négative d’une feature sur une prédiction donnée.

On a choisi de prendre sur la ligne d’indice 0 puis on observe l’impact de chaque feature sur la prédiction qui a été faite pour cet exemple du dataset. Après avoir passé toutes les features, on voit le résultat de 0.29.



L’idée est donc d’imaginer que le modèle va, pour chaque feature en entrée, bouger un slider dans une direction avec une amplitude dépendante de l’importance de la feature. Nous allons faire succéder des glissements du slider, jusqu’à ce que toutes les caractéristiques aient été vues. Les shapes values vont donc être ces valeurs de déplacement du slider. La méthode des shap values consiste à utiliser ces valeurs et à les additionner séquentiellement pour obtenir la prédiction. Le schéma suivant représente les enchaînements sur le slider pour chaque caractéristique :

Une shap value est alors caractérisée par son signe et son amplitude. En effet, le signe permettra d’évaluer l’ordre de corrélation entre la caractéristique et la cible, et l’amplitude donnera l’importance de la feature. La feature la plus importante dans la prédiction sera celle qui aura la shap value la plus grande et positive. La shape value va classer les features des plus importantes au moins importantes.



## **Conclusion :**

En conclusion, ce projet a aboutit à la mise en place d’un modèle de Home Credit Risk Classification en utilisant l’algorithme Random Forest.Grâce à l’utilisation de ML Flow, nous avons pu effectuer un suivi efficace et intuitif de nos expériences, tandis qu eles valeurs Shap ont contribué à une meilleure compréhension de nos prédictions.Nous avons adopté les bonnes pratiques en matière de développement, en utilisant un cookie cutter pour avoir un projet bien structuré et Git pour facilité la collaboration en équipe.